

# A Practical Example for Model-Driven Web Requirements

M.J. Escalona<sup>1</sup>, C.L. Parra<sup>2</sup>, F.M. Martín<sup>3</sup>, J. Nieto<sup>4</sup>, A. Llergo<sup>5</sup>  
and F. Pérez<sup>6</sup>

<sup>1</sup>Department of Computer Languages and Systems, University of Seville, mjescalona@us.es

<sup>2</sup>Hopitales Universitarios Virgen del Rocío, Sevilla, carlos.parra.sspa@juntadeandalucia.es

<sup>3</sup>Servicio Andaluz de Salud, Sevilla, franciscom.martin.sspa@juntadeandalucia.es

<sup>4</sup>Hopitales Universitarios Virgen del Rocío, Sevilla, jaime.nieto.sspa@juntadeandalucia.es

<sup>5</sup>Servicio Andaluz de Salud, Sevilla, antonio.llergo.sspa@juntadeandalucia.es

<sup>6</sup>Servicio Andaluz de Salud, Sevilla, francisco.perez.torres@juntadeandalucia.es

**Abstract** The number of approaches for Web environments has grown very fast in the last years: HDM, OOHDM, and WSDM were among the first, and now a large number can be found in the literature. With the definition of MDA (Model-Driven Architecture) and the acceptance of MDE (Model-Driven Engineering) techniques in this environment, some groups are working in the use of metamodels and transformations to make their approaches more powerful. UWE (UML-Based Web Engineering) or OOWS (Object-Oriented Web Solutions) are only some examples. However, there are few real experiences with Web Engineering in the enterprise environment, and very few real applications of metamodels and MDE techniques. In this chapter the practical experience of a Web Engineering approach, NDT, in a big project developed in Andalusia is presented. Besides, it shows the usability of metamodels in real environments.

## 1 Introduction

In the last years, Web Engineering (Deshpande et al., 2002) has been studied by several important research groups. With the first approach for Hypermedia Systems, HDM (Garzotto et al., 1993), research groups started to propose, develop, and analyze different techniques, models, and procedures in order to offer a suitable methodological environment for the new Software Engineering area.

As a recent development, several groups are basing their approaches on the Model-Driven Engineering (MDE) (Schmidt, 2006) and in the paradigm defined by the OMG with MDA (Model-Driven Architecture) (OMG, 2003). Recent literatures

offer suitable examples of MDE applications in the Web Engineering. They analyze their advantages and their powerful possibilities. However, very few practical applications can be found. Previous comparative studies (Retschitzegger and Schwinger, 2000) (Koch, 1999) (Escalona and Koch, 2004) show that Web Engineering is being ardently studied by the research community, but all of them conclude that it is not applied enough on the enterprise environment. The same is happening with MDE.

This chapter presents a real experience with Web Engineering and MDE. It starts by analyzing the situation of MDE in the Web Engineering environment, and it analyzes briefly its advantages and consequences. In Sect. 3, NDT (Navigational Development Techniques) (Escalona, 2004) is presented. NDT is an MDE approach for Web environments that has been used in different real projects in Spain. NDT is mainly focused on the requirements and analysis phases; for this reason, this chapter presents the practical experience in this area. Although NDT is being applied in several real projects, and several studies about practical experienced have been published (Escalona, 2004) (Escalona et al., 2006), this chapter presents the application of NDT in Diraya project. Diraya was selected for its special characteristics. This is a very ambitious project to manage health information in any hospital in Andalusia. As presented in Sect. 4 Diraya is a complex system with a high group of developers and users; thus, it was an important challenge for our approach. Besides, it requires special aspects: the use of HL7 (<http://www.hl7.org>), the use of a power tool for a large number of developers, etc. that offers an important experience and feedback for our research work. Section 5 shows the real advantages of using MDE and Web Engineering in a project like Diraya. It analyzes how metamodels can offer a powerful environment to fuse approaches, standards or, even, to use different tools based on metamodels. Finally, conclusions and future works are presented.

## **2 Related Works**

Nowadays, MDE and MDA are frequently used. In the Web Engineering environment it is not different. This chapter is mainly focused on the requirements phase; for this reason, we are going to focus on approaches based on metamodels and MDE for Web requirements. However, it is necessary to point out that every day, more Web Engineering research groups are working in MDE environment.

One of the most recent works is (Valderas et al., 2006). This study presents an approach to transform a Web requirements model to a set of prototypes. They propose a requirements treatment based on the task metaphor. Valderas et al. offer an extension of this approach to deal with the specific characteristics of Web requirements. After that, they present a way to derive the navigational model of OOWS (Fons et al., 2003). Firstly, they propose to define requirements like tasks; these tasks are translated into an AGG Graph. Using Graphs transformations, analysis models are obtained. The approach is supported with a tool that is available. This

work is very interesting because they offer a suitable solution for transformation supported by a tool. However, its transformations are not based in OMG tendencies. This shows that they are not compatible with other similar approaches.

In (Escalona and Koch, 2006) the power of metamodel is presented. In comparative studies about Web approaches, a general conclusion is that similar concepts are used or represented with a different number of models, techniques, or artifacts. Thus, for instance, navigational classes are presented with different elements in UWE (Koch, 2001), OOHDM (Rossi, 1996), or WebML (Ceri et al., 2000). Escalona and Koch show in this chapter how a metamodel can represent a concept independent of its representation or notation; only concepts are important. They present a metamodel for Web requirements, named WebRe, that represents requirements models of W2000 (Baresi et al., 2001), NDT, OOHDM, and UWE. Koch et al. (2006) continue their work using QVT to get analysis models from this metamodel. These works are interesting because they are completely based on UML and QVT, standards defined by OMG. However, the results too theoretic.

Fernández and Monzón (2000) present the possibilities of working with metamodels and tools. Thus, they present how a requirements metamodel can be easily defined in IRqA (Integral Requisite Analyzer). IRqA is a commercial tool that lets define metamodels for requirements. In this sense, this study presents the power of tools that support metamodels because they are suitable for any approach defined using metamodels. This work is very practical in fact, but it is not an approach for Web. They do not offer specific artifacts to deal with Web environment; they just offer an approach for classical requirements treatment.

However, although these works are specific for requirements, other classical approaches are working in the MDE environment. For instance in (Schauerhuber et al., 2006) (Moreno et al., 2006), some metamodels for WebML can be found. They present how metamodels can represent classical concepts independent of the artifact used to represent them.

### **3 NDT (Navigational Development Techniques)**

NDT is a methodological approach to deal with requirements in Web Environments. NDT was proposed in order to support the requirements engineering and the analysis phase of Web Systems, and it is based on the MDE paradigm.

Several comparative studies, e.g., that analyzed in (Escalona and Koch, 2004), have proved that one of the less treated phases in Web Engineering is the requirements phase. Most of approaches in Web Engineering are focused on the analysis and design phases. They usually propose to use classical requirements techniques, like use cases, in order to capture and define requirements in Web.

Although use cases is a suitable technique to deal with requirements and it is usually very easy to be understood by the user, frequently they are very ambiguous (Insfrán et al., 2002; Vilain et al., 2000). For this reason, since 2006, several research groups are working in specific requirements treatment for the Web environment. For instance, OOHDM has proposed the UID (User Interaction Diagrams) (Vilain et al., 2000), a specific diagram to deal with interaction requirements.

Another conclusion from comparative studies is that, in Web Engineering, different aspects of software are treated in a separate way. This idea is followed in the analysis and design phases for several approaches, e.g., OOHDM, UWE, WebML, and OOH (Cachero, 2003). This idea of concept separation can be moved to the requirements phase in order to get the advantage of concept separation. Thus, UWE deals separately with information requirements, functional requirements, etc. W2000 defines different use cases for functional and navigational requirements.

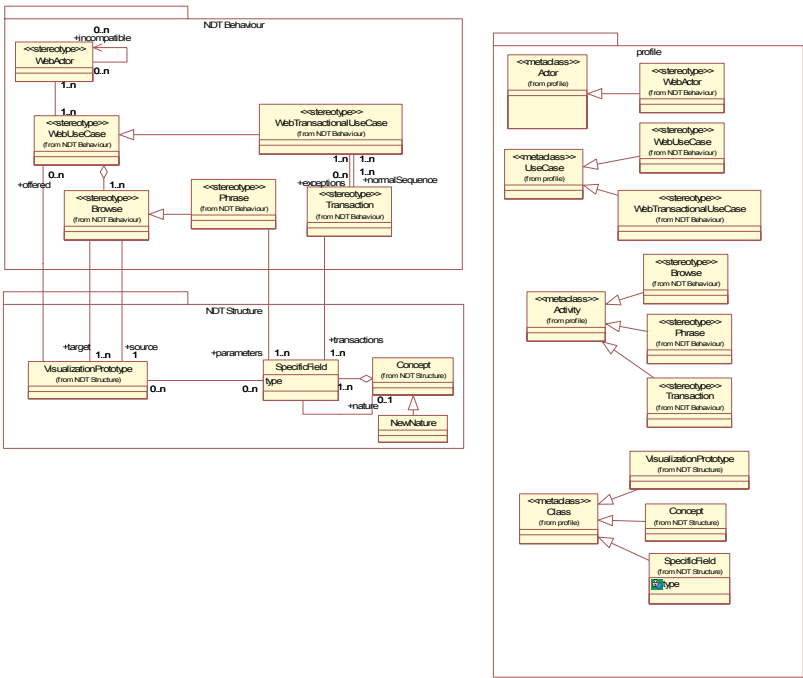


Fig. 1. NDT requirements metamodel

Finally, a fact detected by the comparative studies is that, sometimes, requirements are defined in a very ambiguous way, and it is very difficult for the analysis to translate the knowledge from the requirements definition to the analysis models.

With these ideas, NDT was proposed. Thus, it proposes an MDE approach in order to offer a suitable environment to capture, define, analyze, and validate Web requirements. The life cycle of NDT starts with requirements engineering. Its artifacts are described with a metamodel for requirements that extends the metamodel of UML and follows the structure of WebRe (Escalona and Koch, 2006) meta-model. In Fig. 1 the metamodel of NDT is presented. This metamodel can be defined using a UML profile, which is also presented in Fig. 1. In the requirements level, NDT divides requirements into four different kinds, which are all presented on the metamodel: **storage information requirements**, represented by Concept,

NewNature, and SpecificField metaclasses. They capture the information that the system has to manage. **Actors' requirements**, represented by the WebActor metaclass. A web system can change depending on the user who interacts with it. **Functional requirements**, represented by WebTransactionalUseCase and Transaction metaclasses. They capture information about the functionality that the system has to offer. **Interaction requirements**, represented by Phrases, Browse, and VisualizationPrototype metaclasses that compile the structure of the interaction with the system.

Thus, NDT follows the idea of concept separation. Each of these artifacts is treated with a specific technique. In order to describe each kind of requirements, NDT proposes the use of patterns. A pattern is a table with specific fields for each kind of requirements. Each pattern is a practical vision of each concept in the metamodel. For instance, in Table 1 the specific pattern to describe an actor is presented. This pattern was derived from our practical experience in Diraya project. Although not presented in the metamodel in Fig. 1, each row in the pattern is an attribute of the WebActor metaclass in the metamodel. Thus, patterns are really an easy interface to present the metamodel to the development team. Following the MDA notation, the requirements level of NDT represents the CIM (Computation Independent Model) level of this MDA approach.

**Table 1.** An example of NDT patterns

<b>ACT-01</b>	Health expert
<b>Associated</b>	OBJ-01. Manage information about specialists that work with Diraya
<b>Description</b>	This actor represents any person who interacts with Diraya system. It represents doctors, nurses, janitor, etc.
<b>Comments</b>	This actor assumes the functionality common for any actor in Diraya

The life cycle of NDT continues with the analysis phase. However, in this phase, NDT does not propose any analysis metamodel. As concluded in several comparative studies, there are too many analysis approaches for Web Engineering. For this reason, NDT uses the UWE analysis metamodel (Kraus and Koch, 2003). The selection of UWE is because UWE is completely based on UML, and also, UML profiles are defined. Thus, in this phase, a content model, a navigational model, and an abstract interface model are generated. UWE analysis models represent the PIM level of NDT. Between the CIM and the PIM metamodels, NDT defines a set of relations and transformations using QVT. Thus, NDT solves some problems detected:

It offers specific techniques for each kind of Web requirements.

It proposes the use of an MDE environment in order to make easier the translation between requirements and analysis.

It proposes the use of patterns to deal with requirements. Patterns are very easy to be understood by the user, but they also represent metamodel artifact in a structured way that is very useful for the development team.

It follows the separation concept paradigm proposed by several research groups for the design and the analysis phase. Thus, it can assume the advantage of this paradigm.

Besides, NDT is not only a theoretical approach. It has been applied in several real projects as presented in (Escalona et al., 2006). This study analyzed real experiences of NDT and how they are influenced in the approach. Nowadays, it is applied by several public organizations like Culture Andalusian Government (<http://www.juntadeandalucia.es/cultura/>) and Andalusian Health Service (<http://www.juntadeandalucia.es/servicioandaluzdesalud>), as the requirements methodology for their software projects. Also, some private companies are using it (e.g., everis and Telvent) in several projects.

## **4 An Example: Diraya Project**

Diraya Project is a very ambitious project developed by the Andalusian Health Service (SAS in Spanish). SAS is the government organization in Andalusia that manages hospitals, health centers, and other health public systems. In order to manage the information of patients and health centers, SAS developed Diraya System. Some years ago, the Diraya project was used to manage the primary health attendance in Andalusia. This first module was called Primary Diraya. However, now, it is extended to the specialized health centers as Specialized Diraya.

Specialized Diraya is a Web system to manage all the information about patients who visit any hospital in Andalusia, irrespective of whether they visit the hospital with a prior appointment or as a case of urgency. Specialized Diraya will be installed in 34 hospitals in Andalusia, and it will be used by more than 62,000 final users composed of doctors, nurses, etc.

The project started some months ago, and it is developed by six big software companies in Andalusia: Telvent, Indra, everis, Tecnova, Accenture, and Isoft. They are working together in order to get the best results. They are experts in health systems, and they are using their previous knowledge during the development process. Specialized Diraya development is divided into two phases. The first one was presented in June 2007 and it was installed in one hospital in order to value it. The analysis phase has just been completed.

Specialized Diraya development is based on Métrica (2007), a methodology proposed by the Spanish government for public software projects. Métrica is a very complex and very extensive approach that was used on UML models for modeling systems. NDT is also based on UML; thus, Métrica and NDT are compatible. NDT offers a normalization of Métrica for Web System. For this reason, it was the platform used in the requirements and the analysis phase.

The group of analysts is composed of 13 people from the different companies, and they are working together in order to get a consistent product. The complete development team includes people from companies, analysts, designers, software experts in SAS, etc. It is composed of more than 40 people.

The magnitude of Diraya is a very interesting example. The large number of final users, analysts, designers, etc., the huge number of requirements, and the vast number of different roles in the development offer a complex and an interesting real example of our research results.

## **5 The Practical Experience**

For 6 months approximately, the group of Specialized Diraya was working in the requirements and the analysis phase. In order to use NDT as a methodological environment in these phases, the group had to follow some guidelines.

The first one was to define a suitable architecture for the system. Diraya is a very big project. For this reason, it was decided to divide it into different modules. Thus, six modules interconnecting each other were defined. The division into six modules follows a practical decision. Each company is expert in one area. The system was divided into these areas and each company developed one of them.

Another decision very important for the project was the selection of a tool. In Sect. 5.1 the selection is presented. In the study of available tools, the use of metamodels was an important advantage for the team.

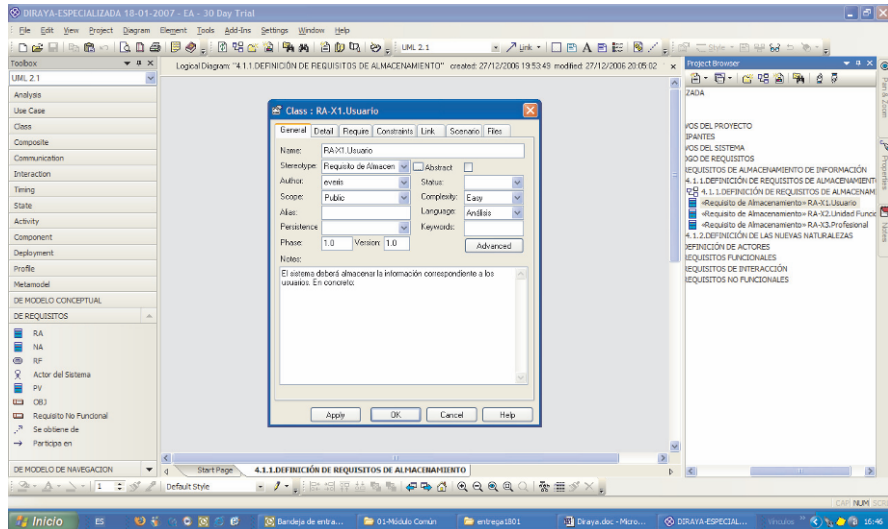
A guideline proposed by SAS is the use of a standard, HL7, for representing the communication of the system. In Sect. 5.2 the power of metamodels for fusing approaches like NDT and HL7 is presented. Finally, in Sect. 5.3, the advantages of using MDE in Diraya are offered.

### ***5.1 The Power of Metamodels for Tools***

Another important decision in the project was the tool. With more than 13 people working in the analysis phase and a large number of requirements, a tool had to be used. To begin with, the first selection was NDT-Tool (Escalona et al., 2003). NDT-Tool covers the complete life cycle of NDT. It was developed by the University of Seville and it is free. The first version of NDT-Tool was developed in Visual Basic. Now, a new version developed in J2EE is being developed in collaboration with two companies, everis and Telvent. When it was tested by Diraya group, the old version was not suitable for a group of analysts that are working in different offices and at the same time. The new version in J2EE via internet was a good choice, but it was too young for the project.

For this reason, SAS and the University of Seville were worked in order to find a suitable tool for Specialized Diraya. After looking at and comparing different possibilities, Enterprise Architecture (<http://www.sparxsystems.com/>) was the selection. Enterprise Architecture was selected for several reasons. It supports UML, and it offers the possibility of extending the initial definition of UML with its extension mechanisms. NDT requirements metamodel and also UWE analysis model are defined with a UML profile, as presented in Sect. 3. For this reason, it was very easy to adapt the tool for the group. With Enterprise Architecture profile option, the profile of NDT was defined, and companies used them to define requirements

and analysis artifacts. Besides, Enterprise Architecture was not a very expensive tool and its interface is very suitable for working. The last point for this selection was the possibility of defining documents normalized according to NDT, UWE, and SAS preconditions in Enterprise Architecture. In Fig. 2 the interface of Enterprise used in Diraya is presented.



**Fig. 2.** Enterprise architecture interface for Diraya

On the left, profiles for each phases of NDT are defined (requirements, conceptual model, and navigational model). In the right an index for NDT results in each phase is presented.

With this environment, companies can work with NDT metamodels without any special knowledge about metamodels. Besides, Enterprise Architecture offers a tool to define transformation between models using MDA and, now, we are trying to implement NDT transformation in order to get the derivation of CIM to PIM model automatically.

However, the power of metamodels for using tools offers a bigger number of tools for using an approach. In our experience other possibilities were offered to the companies. Thus, for the second phase of Diraya, a double use of tool is being adapted. IRqA (<http://www.irqaonline.com/>) is a tool especially for the requirements treatment. It is also based on UML and it offers the possibility of defining metamodels and UML profiles. IRqA offers a better interface for users. Although the phase of requirements will be dealt with IRqA, the rest of the life cycle, even the analysis, will be treated with Enterprise Architecture.

Nowadays, we are working in the generation from IRqA models to the Enterprise Architecture models using the QVT relations of NDT.

Thus, thanks to the definition of metamodel, Diraya users has three possibilities: NDT-Tool, Enterprise Architecture, and the fusion between IRqA and Enterprise Architecture.



In conclusion, if an approach works with UML profiles, its artifacts can be easily defined in tools that work with UML and offer its extensions possibilities. Nowadays, other tools are adapted to NDT. The idea is to offer a different number of possibilities. Thus, they could select the best for their companies and their environment.

## ***5.2 The Power of Metamodels for Merging Approaches***

Another important lesson learned is the power of metamodel for merging approaches. NDT is a methodology that works only in the requirements phase and the analysis. When NDT was proposed, we noticed the importance of making it compatible with other works and approaches. As it was introduced, comparative studies concluded that there were too many models and techniques that dealt with the same concepts in Web Engineering. For this reason, NDT uses UWE metamodels for analysis.

However, although metamodel can be very useful to make compatible different software engineering approaches, in Diraya project we detected that they are also very useful in order to fuse approaches of different environments.

In medical environments, a standard, named HL7, was defined in order to get a standard communication system in medical software system. In Diraya project an important requirement was to follow this standard.

Fortunately, HL7 is defined using also a set of metamodels. It offers a metamodel for use case and interaction metamodel, information model, message design and data type, and vocabulary. HL7 metamodel must be used in the design phase, and it is necessary to translate user requirements into its metamodel artifacts.

In order to make the use of NDT useful for SAS requirements, we studied HL7 and NDT metamodels. Thus, some correspondences between analysis and design concepts were defined.

## ***5.3 The Advantages of MDE***

Although in the first phase of Diraya a practical tool to translate requirements into analysis models was not offered, because the time of this phase was too hard and tools could not be prepared, MDE was still applied.

As it was presented, NDT transformations are defined with QVT. In NDT-Tool, these transformations were translated into Java algorithms. Thus, the step from requirements to analysis is automatic. These algorithms were offered to analysts and they applied them manually. Although this process is not the most orthodox, in the enterprise environment the time of development is one of the most important elements, and they could not wait for the preparation of Enterprise Architecture.

Obviously, for analysts, the manual application of MDE was not easy, but the result was very important for the project. The capture and definition of requirements in Diraya took more than 3 months. During this phase, the results were:

- More than 200 storage information requirements
- Sixty-seven actors defined

- More than 250 functional requirements
- More than 210 interaction requirements

Despite this large number and the manual application of MDE transformation, the generation of the analysis models just took 1 week. This is despite the fact that the coordination group estimated more than 2 months at the beginning of the project when the use of MDE was not included.

This reduction of time was an important advantage detected and pointed out by the companies and SAS.

Besides, during the MDE generation, the consistency between requirements definition and analysis model is assured. MDE keeps the consistency between models, and it assures that requirements and analysis models represent the same. Thus, it was very interesting to detect the high number of failures and mistakes during the transformation. When analysts generated the analysis models from the requirements, they detected that, sometimes, the analysis model that they had in mind was not the same as that represented by their requirements definition.

The consistency that MDE offers, stops the snow ball effect. At the beginning of the life cycle, the correction of mistakes is simple, but in each posterior phase, the correction is more expensive. The cost of changes grows like a snow ball. Thus, the correction of the requirements definition in Diraya just takes 3 days. However, it would be more expensive if they were detected in a posterior phase, for instance, in the implementation phase.

## 6 Conclusions and Future Works

The rapid advance in the Web Engineering research and approaches presented in the last years offers suitable environments to work in Web development for companies and real projects. The advantages included with the application of MDE have improved these environments. However, these environments are not really applied in practical experience. In Web Engineering, and in Model-Driven Web Engineering, there is an important gap between theory and practice.

In the literature, studies have presented the advantages of applying Web Engineering in Web projects, the reduction of time of MDE, the consistency between model when MDE is used, the power of metamodels, etc. However, very few practical experiences in real projects, with real development teams are presented. This chapter offers a different vision of Web Engineering. It shows a practical experience with Web Engineering in a real project. The study has presented NDT, a methodological approach to deal with requirements in web systems based on MDE, and a big project developed in Andalusia, Specialized Diraya.

The research has offered a practical and a real vision about the application of NDT to Diraya. It has presented how metamodels are optimal to fuse approaches and offer a group of suitable tools for the development team. Besides, it presents how MDE can reduce the time of development and detect inconsistency and mistake in early phases, although in our case MDE was applied manually.

Obviously, tools offered by Diraya are not the best ones. At present, we are looking at implementing QVT transformations in order to offer the possibility of automatic generation. In Enterprise Architecture, a language for MDA applications is offered, but it is not based on QVT and it is not sufficiently documented, so we are finding a lot of problems. In any case, we think that it is very important to count with practical approaches like the presented one. They offer an important feedback for research results. For us, the most important learned lessons were:

- Metamodels are a powerful tool for methodological environment. However, development team is not usually an expert in metamodels. For this reason, it is necessary to offer a suitable interface to work with these metamodels. In the case of NDT, it proposes to use patterns. They are very useful to work with the team and also with the final users.
- Metamodels are powerful, but they must be compatible. For instance, for the fusion between NDT and HL7, if both are defined using a UML profile, it would be easier to detect the common artifacts.
- MDE and Web Engineering are necessary because Web systems have special characteristics that must be treated in a special way. However, they are useless in the enterprise environment if they are not supported by a tool. For companies, the development team is an important variable, and if they have to spend a lot of time in documentation, modeling, or transformations, they will never use research approaches.

Commercial tools are supporting the definition of metamodels and profiles. However, they are starting to work with MDE transformations. The research community needs powerful tools to implement transformations. For this reason, one of our future works is researching in tools that have offered these possibilities, like SmartQVT (<http://sourceforge.net/projects/smartqvt>) or Moment (Queralt et al., 2006), and measuring if they are useful for the enterprise environment.

## References

- Baresi, L., Garzotto, F., Paolini, P. (2001). Extending UML for Modeling Web Applications. Proc. HICSS 2001, pp. 1285–1294.
- Cachero, C. (2003). Una extensión a los métodos OO para el modelado y generación automática de interfaces hipermediales. PhD Thesis. Universidad de Alicante, Alicante, España.
- Ceri, S., Fraternali, P., Bongio, A. (2000). Web Modelling Language (WebML): A Modelling Language for Designing Web Sites. Conference WWW9/Computer Networks (33) pp. 137–157.
- Deshpande, Y., Marugesan, S., Ginige, A., Hanse, S., Schawabe, D., Gaedke, M., White, B. (2002). Web Engineering. Journal of Web Engineering, 1(1), 3–17.
- Escalona, M.J. (2004). Modelos y técnicas para la especificación y el análisis de la Navegación en Sistemas Software. Ph. European Thesis. University of Seville, Spain.
- Escalona, M.J., Koch, N. (2004). Requirements Engineering for Web Applications: A Comparative Study. Journal of Web Engineering, 2(3), 193–212.

- Escalona, M.J., Koch, N. (2006). Metamodelling the Requirements of Web Systems. Proc WebIST 2006, pp. 310–317, Setúbal, Portugal, April 2006. ISBN: 978-972-8865-46-7.
- Escalona, M.J., Mejías, M., Torres, J., Reina, A.M. (2003). NDT-Tool: A Tool Case to Deal with Requirements in Web Information Systems. Proc ICWE 2003, LNCS 2722, pp. 212–213.
- Escalona, M.J., Gutierrez, J.J., Villadiego, D., León, A., Torres, A.H. (2006). Practical Experience in Web Engineering. Proc ISD 2006, Budapest, Hungary.
- Fernández, J.L., Monzón, A.A. (2000). Metamodel and a Tool for Software Requirements Management. Reliable Software Technologies. Ada-Europe, Germany.
- Fons, J., Pelechano, V., Albert, M., Pastor, O. (2003). Development of web applications from web enhanced conceptual schemas, ER'03, Springer, LNCS Vol. 2813.
- Garzotto, F., Schwabe, D., Paolini, P. (1993). HDM-A Model Based Approach to Hypermedia Application Design. ACM Transactions Information Systems, 11(1), 1–26.
- Insrán, E., Pastor, O., Wieringa, R. (2002). Requirements Engineering-Based Conceptual Modelling. Requirements Engineering Journal, 7(1).
- Koch, N. (1999). A Comparative Study of Methods for Hypermedia Development. Technical Report 9905. Ludwig-Maximilian-University, Munich, Germany.
- Koch, N. (2001). Software Engineering for Adaptive Hypermedia Applications. Ph. Thesis, FAST Reihe Softwaretechnik, Vol(12), Uni-Druck Publishing Company, Germany.
- Koch, N., Zhang, G., Escalona, M.J. (2006). Model Transformations from Requirements to Web System Design. Proc ICWE 2006, pp. 281–288.
- Kraus, A., Koch, N.A. (2003). Metamodel for UWE. Technical Report 0301, Ludwig-Maximilians-Universität München.
- Métrica v3. (2007). Ministerio de Administraciones Públicas. <http://www.map.es>
- Moreno, N., Fraternali, P., Vallecillo, A.A. (2006). UML 2.0 Profile for WebML Modeling. II International Workshop on Model-Driven Web Engineering. Palo Alto, CA.
- OMG. (2003). MDA Guide, <http://www.omg.org/docs/omg/03-06-01.pdf>. Version 1.0.1.
- Queralt, P., Hoyos, L., Boronat, A., Carsí, J.A., Ramos, I. (2006). Un motor de transformación de modelos con soporte para el lenguaje QVT relations. III Taller sobre Desarrollo de Software Dirigido por Modelos. MDA y Aplicaciones (DSDM'06). España.
- Retschitzegger, W., Schwinger, W. (2000). Towards Modeling of Data Web Applications – A Requirements Perspective. Proc AMCIS 2000, Vol. 1, pp. 149–155.
- Rossi, G. (1996). An Object Oriented Method for Designing Hipermedia Applications. Ph.D. Thesis, Departamento de Informática, PUC-Rio, Brazil.
- Schauerhuber, A., Wimmer, M., Kapsammer, E. (2006). Bridging existing Web Modeling Languages to MDE: A Metamodel for WebML. II International Workshop on Model-Driven Web Engineering. Palo Alto, CA.
- Schmidt, D.C. (2006). Model-Driven Engineering. IEEE Computer, 39(2), 41–47.
- Valderas, P., Pelechano, V., Pastor, O. (2006). A Transformational Approach to Produce Web Application Prototypes from a Web Requirements Model. International Journal of Web Engineering and Technology (IJWET) (1476-1289).
- Vilain, P., Schwabe, D., Sieckenius, C. (2000). A Diagrammatic Tool for Representing User Interaction in UML. Lecture Notes in Computer Science. UML'2000. York, England 2002.